X

Course outline

How to access the portal

Week 1

Week 2

- Lecture 6: VERILOG LANGUAGE FEATURES (PART 1)
- Lecture 7: VERILOG LANGUAGE FEATURES (PART 2)
- Lecture 8: VERILOG LANGUAGE FEATURES (PART 3)
- Lecture 9: VERILOG OPERATORS
- Lecture 10: VERILOG MODELING EXAMPLES
- Lecture 11: VERILOG MODELING EXAMPLES (Contd)
- Week 2 Lecture Material
- Icarus User Guide
- Programming Assignment 1:4-bit adder-subtractor
- **Programming Assignment 2:4-bit S-R latch**
- Programming Assignment 3:8-to-1 Multiplexer
- Quiz : Week 2 Assignment 2
- Feedback for Week 2

Week 3

Programming Assignment 2:4-bit S-R latch

Due on 2018-09-13, 23:59 IST

Write a Verilog module to implement a S-R-type latch at behavioral level. The module will take as arguments the following:

i. 1-bit S and R inputs

ii. 1-bit latch enable input En

iii. 1-bit Q and Qc output

```
module srlatch (S, R, En, Q, Qc);
    input S, R;
    input En;
    output Q, Qc;
```

Write a Verilog module to implement a 4-bit latch, by instantiating four **srlatch** declared above. The module will take as arguments the following:

i. 4-bit S and R inputs for all four S-R-latches

ii. 1-bit latch enable input En

iii. 4-bit Q and Qc outputs from all four S-R-latches

```
module srlatch4 (S, R, En, Q, Qc);
    input [3:0] S, R;
    input En;
    output [3:0] Q, Qc;
```

Please note that all four S-R-latches will activate on the common enable **En** input line.

DO NOT CHANGE THE MODULE(S) NAME AND ORDER OF THE VARIABLE(S) PASSED AS MODULE PARAMETER(S). KEEP THEM INTACT AS MENTIONED IN THE QUESTION.

Sample Test Cases

	Input	Output
Test Case1	Pass: for Q(t) = 0000, Qc(t) = 1111 and S = 0000, R = 0000, En =1, Q(t+1) = 0000, Qc(t+1) =	
Test Case2	Pass: for Q(t) = 0101, Qc(t) = 1010 and S = 1100, R = 0000, En =0, Q(t+1) = 0101, Qc(t+1) =	
Test Case3	Pass: for Q(t) = 1111, Qc(t) = 0000 and S = 1100, R = 0011, En =1, Q(t+1) = 1100, Qc(t+1) =	
Test Case4	Pass: for Q(t) = 1100, Qc(t) = 0011 and S = 1010, R = 0101, En =1, Q(t+1) = 1010, Qc(t+1) =	

Due Date Exceeded.

As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

Select the Language . Verilog ▼

```
1 // Write the srlatch and srlatch4 modules here
2 //*********************************************************************
3 //Sample Solution
4 //*********************************************************************
5 //Single bit S-R latch module
6 module srlatch(S,R,En,Q,Qc);
```

Download Videos

Assignment solution

Text Transcripts

```
    input S, R;
    input En;
    output Q, Qc;
    assign  Q  = ~(Qc | (R & En));
    assign  Qc = ~(Q  | (S & En));

endmodule

//4-bit S-R latch module
module srlatch4(S,R,En,Q,Qc);
    input [3:0] S, R;
    input En;
    output [3:0] Q, Qc;

    srlatch l0 (S[0],R[0],En,Q[0], Qc[0]);
    srlatch l1 (S[1],R[1],En,Q[1], Qc[1]);
    srlatch l2 (S[2],R[2],En,Q[2], Qc[2]);
    srlatch l3 (S[3],R[3],En,Q[3], Qc[3]);
endmodule
//*********************************************************************
//End of Sample Solution
//*********************************************************************
    module srlatch4_tb;
        reg [3:0] S, R, Qp, Qcp;
        reg En;
        wire [3:0] Q, Qc;

        srlatch4 srl4 ( S, R, En, Q, Qc);
            srlatch  srl  (S[0],R[0],En,Q[0], Qc[0]);

        parameter STDIN = 32'h8000_0000;
        integer testid;
        integer ret;

        initial begin

            ret = $fscanf(STDIN,"%d",testid);
            case(testid)
                0:      begin
                        S = 4'b0000; R = 4'b1111; En = 1;
                        #5; {Qp, Qcp} = {Q, Qc}; S = 4'b0000; R = 4'b0000; En = 1;
                    end
                1:      begin
                        S = 4'b0101; R = 4'b1010; En = 1;
                        #5; {Qp, Qcp} = {Q, Qc}; S = 4'b1100; R = 4'b0000; En = 0;
                    end
                2:      begin
                        S = 4'b1111; R = 4'b0000; En = 1;
                        #5; {Qp, Qcp} = {Q, Qc}; S = 4'b1100; R = 4'b0011; En = 1;
                    end
                3:      begin
                        S = 4'b1100; R = 4'b0011; En = 1;
                        #5; {Qp, Qcp} = {Q, Qc}; S = 4'b1010; R = 4'b0101; En = 1;
                    end

                default:    begin
                            $display("Bad testcase id %d",testid);
                                $finish();
                            end
            endcase
                #5;
            if ( (testid==0 && Q == 4'b0000 && Qc == 4'b1111) || (testid==1 && Q == 4'b0101 &&
                            || (testid==2 && Q == 4'b1100 && Qc == 4'b0011) || (testid==3 && Q
                begin
                    if ( ((testid==0 || testid==2 || testid==3) && Q[0] == 1'b0 && Qc[0] == 1'
                        pass();
                    else
                        fail_single();
                end
            else
                fail();
        end

        task fail_single;   begin
                            $display("Fail: for Q(t) = %b,  Qc(t) = %b and S = %b, R = %b, En =%b,
                        $finish();
                    end
        endtask


        task fail;  begin
                            $display("Fail: for Q(t) = %b,  Qc(t) = %b and S = %b, R = %b, En =%b,
                        $finish();
                    end
        endtask

        task pass;  begin
                            $display("Pass: for Q(t) = %b,  Qc(t) = %b and S = %b, R = %b, En
                        $finish();
                    end
        endtask


endmodule

```

End